Especificación Formal de Desarrollo de un Sistema de Firma Digital

Marcelo Daniele
madaniele@gmail.com
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 – San Luis – Argentina
Tel: + 54 (0) 2652 – 424027 ext. 251
Fax: + 54 (0) 2652 – 430059

Resumen

En el presente trabajo se muestra una propuesta de construcción de un software de Firma Digital valiéndose de métodos formales de especificación y desarrollo.

Se utiliza para la construcción de la propuesta el Lenguaje de Especificación RAISE (RSL), que es un lenguaje modular y formal, con una fuerte base matemática, suficiente como para soportar la definición precisa de requerimientos de software y un verdadero desarrollo de definiciones propias para implementaciones ejecutables.

Se comienza por establecer una introducción a la tecnología de Firma Digital, y el estado actual de su utilización en la región, y aspectos legales que se desprenden de ella.

La propuesta continúa por introducir las herramientas matemáticas relacionadas con la Firma Digital, mencionando conceptos de criptografía y criptoanálisis.

El trabajo concluye con una propuesta basada en los requerimientos legales y específicos a Firma Digital, definiendo una arquitectura de construcción formal en lenguaje RAISE.

Con este trabajo se pretende que el lector tenga una idea del estado actual del arte con respecto a la tecnología de Firma Digital y comprenda la importancia que ésta posee en la protección de documentación electrónica.

Palabras Clave: Firma Digital, Ciptosistemas, Criptoanálisis, RAISE, RSL.

1. Introducción

Desde las civilizaciones más antiguas, el hombre ha tenido la necesidad de comunicarse con los demás y ha trabajado muy duramente para lograr formas cada vez más sofisticadas de realizar este cometido. Con el auge de la comunicación y a causa de éste surgen nuevas necesidades: establecer mecanismos de ocultación de la información que se comparte, mecanismos de privacidad, es decir, formas de que dicha información sólo pueda ser entendida por su destinatario, y además mecanismos para asegurar que el contenido de los mensajes no pudieran ser alterados y si es así que el cambio fuese detectado. Se ha trabajado mucho sobre estos temas, sin embargo estas características aun no se han implementado en tantos sistemas como habría de esperarse.

El objetivo de la Firma Digital no es menos que el de proporcionar la misma, o quizá más aún, seguridad, confiabilidad y legalidad que la firma manuscrita brinda a un documento en papel; sin duda la diferencia principal entre los dos tipos de firma reside en que, para Firma Digital, la información que es firmada se presenta como Documentos Digitales que son transportados por medios electrónicos (Internet, medios de almacenamiento de todo tipo, redes locales, etc.) desde su autor (el firmante) hasta el destinatario (el verificador de la firma).

La Firma Digital es un mecanismo que asegura la **integridad** de documentos almacenados digitalmente. Permite corroborar que el contenido de estos documentos no ha sido alterado, valiéndose de métodos matemáticos relacionados con la criptografía.

Una segunda característica importante de la Firma Digital es que permite asegurar la **pertenencia** de un documento digital a su emisor o persona firmante, es decir, quien recibe el documento digital firmado con el mecanismo de Firma Digital, podrá estar seguro de la autenticidad del mismo.

Esta segunda característica da cabida a una tercera: el **no repudio**. No repudio significa que quien recibe y verifica con éxito un documento digital no está en el derecho de devolver o repudiar el documento a su emisor, aludiendo falsedad en su autenticidad o manipulación del mismo.

2. Estado actual de la tecnología de Firma Digital

En muchos países la Firma Digital ha logrado introducirse con vital importancia en diversos sectores de la sociedad. Se ha logrado su aplicación en circuitos administrativos del Estado, documentos de identidad digitales, voto electrónico, y en una diversidad de documentos que requieran de Firma Digital.

En la República Argentina, la Firma Digital está en proceso de evolución, aún no se cuenta con aplicaciones específicas. Sin embargo existe, ya desde hace un par de años, legislación para apoyar los aspectos legales de la Firma Digital.

En 1998 el artículo 1º del Decreto 427 aprueba la infraestructura de firma digital para la Administración Pública Nacional y faculta a la Jefatura de Gabinete de Ministros como autoridad de aplicación, además fija los estándares (resolución 194/98) a ser utilizados para tal fin.

Luego surgen como importante la ley 25506 del 14 de noviembre del 2001 [1] que establece la ley para Firma Digital, y el decreto 2628 del 19 de diciembre del 2002 que reglamenta la ley vigente.

3. Mecanismos de Firma Digital

El mecanismo que permite firmar digitalmente un documento puede describirse de la siguiente manera [2]: un usuario del software de Firma Digital, la persona firmante, posee una llave digital (también llamada clave o firma), que no es más que un bloque de datos compuesto de dos partes: una parte pública y una parte privada. Este usuario procede a utilizar su software de Firma Digital conjuntamente con la parte privada de su llave digital para aplicar el proceso matemático de firmado al documento digital que el usuario necesite firmar. El resultado de este proceso es el documento digital firmado y un **Certificado Digital** en donde consta la firma del documento y la parte pública de la llave digital, entre otras cosas (Figura 1).

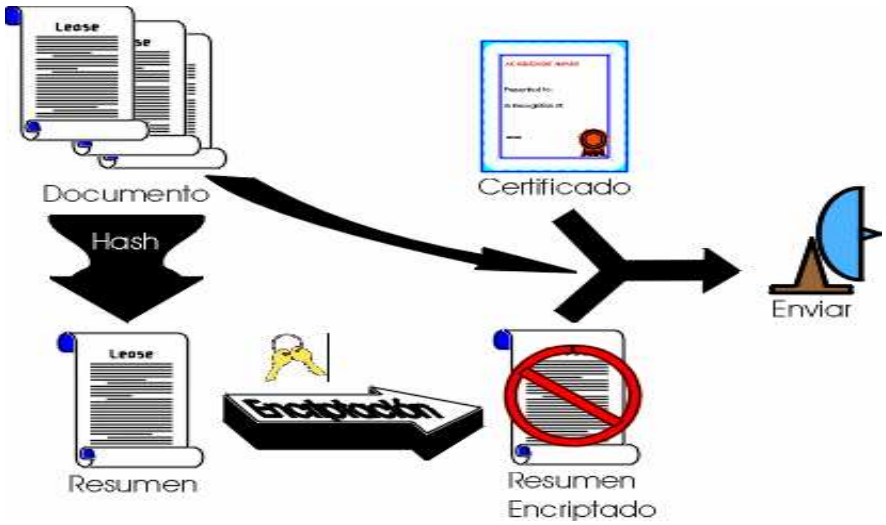

Figura 1. Proceso de firmado de un Documento Digital

Por otro lado el usuario verificador, el receptor del documento firmado, procederá a la verificación del mismo, utilizando la parte pública de la llave digital del emisor y el Certificado Digital recibido (Figura 2). El resultado de la verificación es la autenticidad del documento.

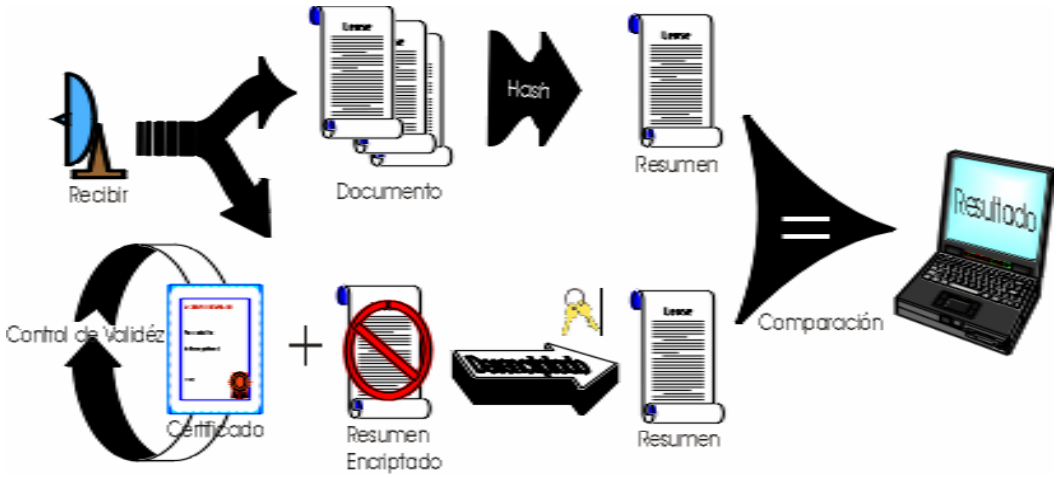

Figura 2. Proceso de verificación de un Documento Digital Firmado

Un usuario firmante necesita requerir al Ente Certificante una autorización para firmar digitalmente, esta autorización es un Certificado Digital, firmado digitalmente

por la Entidad Certificante, sujeto de verificación en el momento de probar la autenticidad de un Documento Digital firmado por éste usuario. El requerimiento de un Certificado Digital puede verse en la Figura 3.

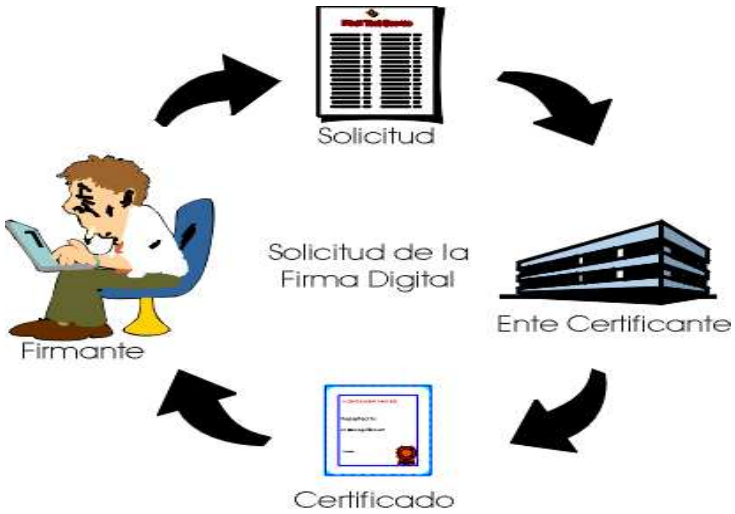

Figura 3. Circuito de solicitud de una Firma Digital

El Certificado Digital es un documento legal que certifica la firma del emisor. Éste se encuentra firmado digitalmente por una Entidad Certificante, por tanto cuenta con el respaldo legal pertinente.

La Entidad Certificante también cuenta con su propio Certificado Digital, permitiendo verificar la autenticidad de los Certificados Digitales emitidos por un usuario firmante que posea autorización para firmar. Esto cierra una cadena de firmas que van desde una autoridad mayor de firmas y desciende hasta el usuario final. La jerarquía con la cuál se rigen las Entidades Certificantes puede verse en la Figura 4.

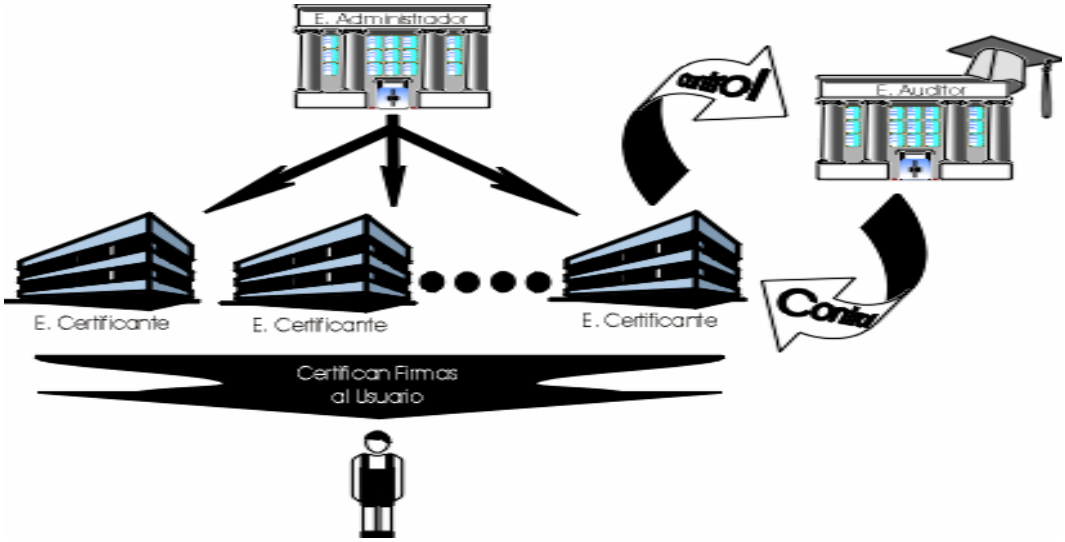

Figura 4. Jerarquía de Entidades Certificantes

4. Criptografía y Criptoanálisis

La criptografía es el arte de enmascarar los mensajes con signos convencionales, que sólo cobran sentido a la luz de una clave secreta. Es la ciencia y el arte de escribir para que sea indescifrable el contenido del texto escrito, para quien no posea la clave. La criptografía actual encuentra su fortaleza en la seguridad de sus claves y no en el secreto de su método de ocultación de los mensajes, generalmente el método es público y conocido.

En un principio, la criptografía era considerada un arte. Pero actualmente es considerada una ciencia aplicada, una rama de las matemáticas, debido a su relación con otras ciencias como la teoría de números, la estadística, teorías de la información y de información computacional.

Criptosistema, es el conjunto de pasos o procedimiento por el cual se alcanza un mensaje cifrado, o se obtiene la firma de un mensaje, y además describe los pasos inversos por los cuales se obtiene el mensaje en claro o, para Firma Digital, la verificación de los mensajes firmados.

Si el propósito de la criptografía es garantizar la seguridad y el secreto de un mensaje, el criptoanálisis busca por todos los medios descubrir ese mensaje secreto, o la clave con la que está codificado, lo que permitiría entender todos los mensajes posteriores.

El criptoanálisis, es la contrapartida de la criptografía. Ambos han tenido una gran relevancia en la historia y en la actualidad, han cambiado el rumbo de las guerras. Juntos, criptografía y criptoanálisis, constituyen la criptología. La lucha entre ambas ciencias, puede equipararse con la metáfora del escudo y la espada: a lo largo de la historia, la criptografía ha desarrollado criptosistemas aparentemente indescifrables, que con el tiempo han sido vencidos por nuevos métodos de criptoanálisis. Puede decirse que ambos evolucionan a la par.

La criptografía utilizada en Firma Digital se vale de un conjunto de algoritmos matemáticos que se mencionan a continuación.

4.1 Algoritmos de Hash o Resumen:

Los algoritmos de hash representan funciones de un solo sentido, o sea, es sencillo tomar un bloque de datos (expresado en binario) y obtener un resumen o digesto siguiendo algún mecanismo de “selección” de componentes, pero imposible (o debería serlo) obtener el bloque completo a partir del resumen. Aquí descansa la preservación de integridad propuesta por la Firma Digital.

Una característica importante del algoritmo es que dado un mensaje de tamaño arbitrario, produce una salida de tamaño fijo, generalmente de 128 o 160 bits. Pero esto trae aparejado un problema bastante grande: dado que la cardinalidad del espacio de todos los mensajes posibles es mucho mayor que el número de combinaciones distintas para un tamaño determinado de hash, necesariamente existen diversos mensajes que producen el mismo resultado, aunque es **computacionalmente imposible** encontrarlos.

Es de todas maneras necesario que el método evite (en la medida posible) colisiones en los resúmenes generados, pues no resulta útil si se logra el mismo digesto desde dos bloques distintos.

Se destacan algunos algoritmos utilizables que fueron o son estándares fijados por algún organismo afín, entre ellos SHA1, SHA2, MD (desde el 1 hasta el 5), RIPEMD-128, RIPEMD-160.

4.2 Algoritmos Asimétricos o de clave pública

La filosofía predominante en un criptosistema de clave pública es, a través de aplicar un proceso de "ocultamiento" utilizando parámetros que involucran datos privados del usuario, proteger el digesto de un bloque de datos agregando identidad a un mensaje basado en estos datos. Se provee a un usuario firmante de un par de claves, una pública y una privada, relacionadas entre sí de manera que un bloque encriptado con una clave (en firma digital la privada) pueda ser desencriptado con la otra (la pública). Estas claves son provistas por algoritmos afines, que varían según la necesidad del método utilizado.

Todos los algoritmos útiles de clave pública aseguran la imposibilidad de averiguar la clave privada de una persona teniendo solo la pública, esta afirmación se apoya en el **problema de logaritmos discretos** (aprovechado por algoritmos criptográficos como: ElGamal, RSA, DSA y ECDSA), explicar la complejidad del problema de logaritmos discretos se escapa de los objetivos y alcances de este trabajo.

4.3 Algoritmos Simétricos

Utilizado para proteger información crucial en el Sistema de Firma Digital, como lo es la clave privada del firmante. Aquí a diferencia de los sistemas asimétricos o de clave pública se puede, en base a una única información conocida por el usuario (no necesita pares de claves, utiliza una única clave) encriptar y desencriptar a discreción la información privada que posee.

Existen numerosos algoritmos de tipo simétrico, se destacan: el DES, triple DES, IDEA.

4.4 Algoritmos de generación de pares de llaves

Como el espíritu del criptosistema de clave pública radica en la utilización de números primos relativos es necesario contar con un mecanismo suficientemente bueno para permitir la selección aleatoria de un número primo que contenga los suficientes bits como para proveer la seguridad esperada.

Para el método RSA existen algoritmos que comprueban la primalidad de números generados con algún tipo de criterio. La mayoría de estos test son no determinísticos, pero se ha avanzado en la construcción de algoritmos determinísticos desde hace poco tiempo; aunque para la variante con curvas elípticas del DSA ya es determinístico desde su concepción.

Como principal componente la generación de números primos necesita valerse de algoritmos de test de primalidad.

Un test de primalidad interesante es el de Rabin-Miller.

5. Propuesta de construcción formal de un software de Firma Digital

5.1 Introducción a RAISE

El Lenguaje de Especificación RAISE (RSL) es un lenguaje modular y formal, con una fuerte base matemática, suficiente como para soportar la definición precisa de requerimientos de software y un verdadero desarrollo de definiciones propias para implementaciones ejecutables. Con características fuertes que permiten la especificación de programas grandes y modulares, diferentes estilos de especificación como (axiomático y basado en modelo, aplicativo e imperativo, secuencial y concurrente) y soportar especificaciones que van desde lo abstracto (cerrado en requerimientos) hasta lo concreto (cerrado en implementaciones) [3].

El método RAISE es la guía para el desarrollo en lenguaje RSL, éste método se basa en cuatro principios [4]: el desarrollo separado, el desarrollo paso a paso, inventar y verificar y por último el desarrollo riguroso.

Una completa referencia sobre el método RAISE y el lenguaje RSL puede obtenerse desde el sitio de UNU-IIST (International Institute for Software Technology de la United Nations University) [5].

5.2 Tipos Básicos

El primer paso en la especificación desde lo formal para la propuesta de software de Firma Digital consiste en definir los tipos básicos a ser considerados.

Así la primera clase lucirá de la siguiente manera (RSL-Código 1):

```
/*Clase base para especificar los tipos Byte, Cardinal y Big como subrango de enteros solo
que con distintos tamaños de representación*/
scheme TBase = class
        type
        Byte     = {| n : Int :- n >= 0 /\ n <= (2**8)-1 |},  /*subrango de entero de 8 bits*/
    Cardinal = {| n : Int :- n >= 0 /\ n <= (2**32)-1|},     /*subrango entero de 32 bits */
        Big = {| n : Int :- n>0 /\ n <= (2**2048)-1 |}       /*numero grande de 2048 bits*/
end
```

RSL-Código 1: Definición de tipos Básicos

5.3 Estructura de un Documento Digital Firmado

Esta clase define un objeto capaz de contener la información que luego se volcará en algún medio de almacenamiento. El Certificado Digital que acompaña al Documento Digital Firmado, es el contenedor que portara la clave pública del Emisor. Para esta propuesta se asumirá que es un bloque de datos apropiadamente dispuesto, siguiendo la estructura del estándar X.509.

Existen también en la clase, operaciones que manipulan estos objetos y todas sus partes.

Un documento Digital firmado consta de 3 bloques (RSL-Código 2):

- El bloque de encabezado del documento digital, no es más que el contenedor que describe el estado y las dimensiones del bloque de información y el bloque de firma.
- El bloque de información es el Documento Digital propiamente dicho almacenado dentro del Documento Digital firmado.

- El bloque de firma contiene el resultado de aplicar las operaciones de firmado a través de algoritmos criptográficos de clave pública al digesto del Documento Digital.

```
object SignedFile :
 with TBase in
 class
  type
   SigFile ::
    file : File.File ↔ replace_file
          signature: SignatureStream ↔ replace_signature
    certificate :
         CertRep.Certificate ↔ replace_certificate /*certificate for signed file*/
         value

         update_file: File.File ≍SigFile -~→ SigFile,
   update_signature:  SignatureStream ≍SigFile -~→ SigFile,
         update_certificate: CertRep.Certificate ≍SigFile -~→ SigFile,
         get_content: SigFile → FileStream,
         get_certificate: SigFile → CertRep.Certificate,
         get_signature: SigFile → SignatureStream,

consistent: SigFile → Bool,
         consistent(sf) is
            File.consistent(file(sf)) /\ CertRep.consistent(certificate(sf))
end
```

RSL-Código2: Estructura del objeto SignedFile

La estructura del encabezado en conjunto con el bloque de información están especificados en el objeto File que se describe en RSL-Código 3.

El Certificado Digital esta representado formalmente en el tipo CertRep, definido en CertRep que será explicado más adelante.

Las operaciones de manejo de todos los objetos se han detallado sólo a nivel de signatura. Sin embargo es apropiado conocer el estado de consistencia del sistema de Firma Digital, para ello se define la función consistent, que debe chequear la consistencia del registro file y certificado para asegurar que el Documento Digital Firmado sea consistente.

```
object File :
 with TBase in
 class
  type

    File ::
     header : FileHeader ↔ replace_header
     content :
       FileStream ↔ replace_content /*file content*/

  value
    update_header : FileHeader ≍File -~→ File
    update_content : FileStream ≍File -~→ File
    get_content : File → FileStream
    consistent : File → Bool
    consistent(c) is true
 end
```

RSL-Código3: Estructura del objeto File

El Documento Digital no Firmado o Documento Digital Original puede simplificarse a la definición de dos bloques simples: el bloque de encabezado y el bloque de información, que son idénticos a los definidos para la clase anterior.

5.4 Estructura del Certificado Digital

El Certificado Digital es la guía que ayudara al software de Verificación de Documentos Firmados Digitalmente a dilucidar la consistencia y validez de estos documentos. En esta propuesta de construcción se ha considerado utilizar una estructura compatible con la recomendación X.509 de la ITU (International Telecommunication Union), que es muy ampliamente usada en la mayoría del software relacionado a la Firma Digital.

```
/*The Certificate (X.509 compatible)*/
TBase, CertMetadata

object CertRep :
 with TBase in
 class
  type
   CertificateInfo = CertMetadata.Metadata,
   Certificate ::
    certinfo : CertificateInfo ↔ replace_certinfo
    signature : SignatureStream ↔ replace_signature
  value
   update_certinfo : CertificateInfo ×Certificate -~→ Certificate
   update_signature : SignatureStream ×Certificate -~→ Certificate
   get_signature : Certificate → SignatureStream
   get_public_key : Certificate → KeyStream
   get_serialized_info : Certificate → FileStream,
   consistent : Certificate → Bool
   consistent(c) is
    CertMetadata.consistent(certinfo(c)) /\
    len signature(c) >= 0
 end
```

RSL-Código4: Estructura del objeto CertRep

El Certificado Digital es un documento firmado por la Entidad Certificante que representa la identidad y veracidad de la clave pública del Emisor del Documento. Así quién tenga la clave pública del Ente Certificante, podrá verificar la integridad y validez del Documento Digital valiéndose de la previa verificación del Certificado Digital del Emisor.

La totalidad del Certificado Digital se divide en dos bloques importantes: el bloque de información del Certificado propiamente dicho y el bloque de Firma de la Entidad Certificante que representa la Firma Digital del Certificado (RSL-Código4).

El bloque de Información contiene los datos necesarios para la identificación del mismo certificado, reflejado en los dos primeros campos Versión y Número de Serie).

El siguiente de los campos, el identificador del algoritmo de firma, establece el algoritmo a utilizar para poder verificar el Certificado.

La identificación de la entidad emisora es otro de los campos que establece esta recomendación, el formato de este campo esta establecido por otra recomendación de la ITU, la X.500.

El período de validez del Certificado es uno de los campos importantes, ya que determina el rango entre fechas en que el Certificado es útil, quedando inhabilitado para ser utilizado si el valor de la fecha no esta dentro de este rango. Esta es una característica importante, ya que la ley que rige la Infraestructura de Firma Digital establece la posible cesación de utilización de Firma Digital para cualquier entidad o persona.

```
/*The Certificate Metadata (X.509 compatible)*/
TBase

object CertMetadata :
  with TBase in
  class
    type
       Metadata::
         version: Version ↔ replace_version /*cert version*/
         serialnum: Serial  ↔ replace_serial /*issuer-specific serial number*/
               algid: AlgId ↔ replace_algid /*algorithm identificator*/
               issuer: Person ↔ replace_issuer /* the issuer */
               validity: Validity ↔ replace_validity /*cert validity */
               subject: Person ↔ replace_subject /* the subject */
               subject_pub_key: KeyStream ↔ replace_subject_pub_key
value
      consistent: Metadata → Bool
           consistent(m) is
              len subject_pub_key(m)>= 0
end
```

RSL-Código5: Estructura del objeto CertMetadata

La información de la "metadata" de un Certificado Digital puede verse en el código RSL-Código5.

5.5 El proceso de Firma de un Documento Digital

El proceso de firmar un Documento Digital puede verse en el código RSL-Código6.

La función parcial sign recibe cinco parámetros: una cadena de datos o archivo a firmar, representada por el tipo File; un identificador de la clave privada a utilizar, de tipo Key_id; el usuario y la contraseña para obtener la clave privada desencriptada; y por último el sistema propiamente dicho. La función se vale de unos cuantos pasos intermedios que permiten llegar a un resultado del tipo SigFile, es decir una cadena de datos o archivo Firmado Digitalmente. El primer paso es determinar el contenido del Documento Digital, es decir separado de su cabecera, a través de la función get_content.

```
value
    /*use case: Sign Digital Document*/
sign :
      File.File ≍ Key_Id ≍ User ≍ Password ≍ Sys -~→
        SignedFile.SigFile

sign(fin, kid, u, p, s) is
      let
        content = File.get_content(fin),
        private_key = RepositoryKeys.get(kid, priv_keys(s)),
        private_dec_key = KeyRep.decode_priv_key(p, private_key),
        hash = THash.mk_Hash(content),
        sign = TAsymmetricAlg.encode(hash, private_dec_key),
        cid = KeyRep.certid(private_key),
        certificate = RepositoryCert.get(cid, certificates(s))
      in
         SignedFile.mk_SigFile(fin, sign, certificate)
     end
pre
      kid isin RepositoryKeys.get_by_owner(u, priv_keys(s)),
```

RSL-Código6: Proceso de Firma de un Documento Digital

El siguiente paso obtiene la clave privada encriptada desde el repositorio de claves privadas, valiéndose del identificador obtenido como parámetro y la función get del objeto RepositoryKeys.

Seguidamente, se procede a desencriptar esta clave privada con mecanismos simétricos definidos por el algoritmo utilizado para tal fin, valiéndose del objeto KeyRep y la función decode_priv_key definida en él.

El siguiente paso realiza, basándose de los datos anteriores, el firmado del contenido del Documento Digital, obteniendo como resultado el bloque de datos que representa la firma del resumen del certificado, para esto se utiliza la función encode del objeto TAsymmetricAlg.

Por último se procede a obtener la copia de un Certificado Digital desde el repositorio valiéndose de la clave privada como identificación y construir por fin el Documento Digital Firmado.

Como se aseguró en párrafos anteriores, la función sign es parcial, esto se debe a que se requiere la precondición sobre los datos de entrada siguiente: el identificador de la clave privada debe efectivamente ser uno de los identificadores que comprenden el conjunto de claves que pertenecen al usuario que esta firmando un Documento Digital.

La función mk_Hash es el constructor de un objeto de tipo Hash, que es el resumen del contenido del Documento Digital y será la entrada para el algoritmo de codificación RSA que obtendrá un bloque de datos conteniendo la Firma Digital del Documento.

5.6 El proceso de verificación de un Documento Digital

El sistema debe ser capaz de entregar un resultado verdadero si se verifica que el Documento conserva sus propiedades intactas, es decir, no se ha modificado su contenido. El siguiente código RSL-Código7 muestra esta función del sistema.

La función parcial verify recibe como entrada tres argumentos: el archivo firmado, un identificador que ayudara a encontrar el Certificado Digital de la Entidad Certificante en el repositorio de certificados del sistema y el último parámetro es el sistema mismo. Esta función debe retornar un par de valores: el Documento Digital Original, y un valor de verdad que indicará si la verificación ha sido exitosa, tanto para el mismo Documento digital, como para el Certificado del emisor contra la clave pública de la Entidad Certificante.

Se detallan una serie de pasos intermedios antes de conseguir el resultado mencionado.

En primer lugar, se obtiene el contenido del Documento firmado a través de la función get_content del objeto SignedFile.

Otro objeto importante es el Certificado Digital adjunto al Documento, este es obtenido a través de la función get_certificate también perteneciente al objeto SignedFile. De este Certificado es necesario obtener la clave pública del emisor, utilizando la función get_public_key de CertRep, también se obtendrá el bloque de firma adjunto al Certificado que corresponde al proceso de firma aplicado por el Ente Certificante y será necesario verificar. Este bloque lo retorna get_signature desde el objeto CertRep .

Del mismo modo también es necesario obtener el bloque de firma desde el Documento Firmado, paso realizado por la función get_signature de SignedFile.

El ultimo de los objetos recuperados, necesario para el proceso de verificación, será la llave pública del Ente Certificante, proveniente del Certificado Digital del mismo ente, el Certificado es obtenido desde el repositorio a través del identificador de certificado utilizando la función get de RepositoryCert y la llave pública será devuelta por la función get_public_key que ya ha servido anteriormente.

El primer objetivo es verificar la validez de los Certificados Digitales adjuntos, tanto del que representa al Ente Certificante como el que representa al emisor. Conceptualmente un Certificado Digital es valido, si su fecha de emisión no supera a la fecha actual y la fecha actual no supera a la fecha de expiración del mismo. Este rango de validez, es comprobado por la función is_valid del objeto CertRep.

```
value
/*use case: Verify Signed Digital Document*/
    verify :
      SignedFile.SigFile ×Cert_Id ×Sys -~→
        Bool ×File.File

    verify(fin, cid, s) is
    let
        content = SignedFile.get_content(fin),
        certificate = SignedFile.get_certificate(fin),
        pub_key = CertRep.get_public_key(certificate),
        signature = SignedFile.get_signature(fin),
        cert_signature = CertRep.get_signature(certificate),
        auth_certificate = RepositoryCert.get(cid, certificates(s)),
        auth_pubkey = CertRep.get_public_key(auth_certificate),
        today = TBase.get_ToDay(),
        result = CertRep.is_valid(auth_certificate, today) /\
                  CertRep.is_valid(certificate, today) /\
                 TAsymmetricAlg.verify(
                        CertRep.get_serialized_info(certificate),
                        cert_signature, auth_pubkey) /\
                 TAsymmetricAlg.verify(
                        content, signature, pub_key),
        file = SignedFile.file(fin)
   in
        (result, file)
end
            pre RepositoryCert.is_in(cid,certificates(s)),
```

RSL-Código7: Proceso de Verificación de un Documento Digital Firmado

La última línea antes de retornar el resultado de la función verify del código RSL-Codigo7 se encarga de reconstruir el Documento Digital original, que será retornado junto con el resultado de evaluar los cuatro puntos anteriores, recapitulando: la validez del certificado Digital de la Entidad Certificante, la validez del Certificado Digital del emisor, la verificación de la firma del Ente Certificante sobre el Certificado del emisor y por último la verificación de la firma del emisor sobre el Documento Digital en cuestión.

6. Conclusiones

La especificación de software con métodos formales no es tarea sencilla y no en todos los casos es económicamente adecuado en la relación costo/beneficio. Este trabajo sugiere una prudente construcción de software siguiendo estándares y normas regulatorias para la actividad de Firma Digital.

La utilización de métodos formales ha sido especialmente provechosa, sobre todo por la facilidad con que el método puede refinarse para lograr especificaciones correctas y en pocos pasos, sin la necesidad de la vuelta atrás que provoca la reescritura y rediseño de otros métodos, asegurando que se mantienen los requerimientos desde el principio hasta el final. La posibilidad de utilizar componentes supuestos resueltos en pasos más concretos, hace que se logre rápidamente llegar a una vista global del sistema.

Por otra parte la utilización de la Firma Digital como herramienta crítica para validar la veracidad de la información compartida a través de medios informáticos o electrónicos aporta un gran valor agregado, sobre todo por el creciente interés de las personas por utilizar medios informáticos y electrónicos como principal forma de comunicación. Interés que está fuertemente potenciado por la aplicación de tecnologías de seguridad y confidencialidad como las estudiadas en este trabajo.

Se ha comprobado el gran valor de la criptografía en la sociedad actual y en la historia. En nuestro mundo actual son una constante: Internet, comunicaciones, cuestiones económicas y secretos de Estado que dependen de ella. Incluso se puede afirmar sin temor a equivocarse que han influido en el curso de nuestra historia, en ámbitos tan variados como la economía, las finanzas, la defensa, la política.

La criptografía, toda un área de interés basado en la ciencia matemática, aprovecha su relación con otras áreas del conocimiento como la teoría de la información y de la comunicación, etc., para mejorarse a sí misma.

Teniendo en cuenta la carencia de Entidades Certificadoras y el pobre despliegue de aplicaciones que ejecuten Firma Digital tanto en los circuitos estatales como en las pequeñas y medianas empresas del sector privado de nuestro país, constituye un llamativo nicho para quien busca inversiones rentables, ya sea perfeccionando el negocio de la Certificación o la creación de herramientas de soporte para la Firma Digital.

7. Referencias

[1] “Ley 25.506 de Firmas Digitales” sancionada por el Senado y Cámara de Diputados de la Nación Argentina disponible en www.pki.gov.ar.

[2] “Estándares sobre tecnología de Firma digital para la Administración Pública Nacional” Resolución 194/98, disponible en www.pki.gov.ar.

[3] George, C., Haff, P., Havelund, K., Haxthausen, A., Milne, R., Nielsen, C., Prehn, S. and Ritter, K. “The RAISE Specification Language”, Prentice Hall, UK, 2002.

[4] George, C., Haxthausen, A., Hughes, S., Milne, R., Prehn, S. and Pedersen, J., “The RAISE Development Method”, Prentice Hall, UK, 1995.

[5] RAISE - Rigorous Approach to Industrial Software Engineering (UNU-IIST) www.iist.unu.edu/www/raise/index.html